An Empirical Evaluation of LFG-DOP

Rens Bod
Informatics Research Institute, University of Leeds, Leeds LS2 9JT, &
Institute for Logic, Language and Computation, University of Amsterdam
rens@scs.leeds.ac.uk

Abstract

This paper presents an empirical assessment of the LFG-DOP model introduced by Bod & Kaplan (1998). The parser we describe uses fragments from LFG-annotated sentences to parse new sentences and Monte Carlo techniques to compute the most probable parse. While our main goal is to test Bod & Kaplan's model, we will also test a version of LFG-DOP which treats generalized fragments as previously unseen events. Experiments with the Verbmobil and Homecentre corpora show that our version of LFG-DOP outperforms Bod & Kaplan's model, and that LFG's functional information improves the parse accuracy of tree structures.

1 Introduction

We present an empirical evaluation of the LFG-DOP model introduced by Bod & Kaplan (1998). LFG-DOP is a Data-Oriented Parsing (DOP) model (Bod 1993, 98) based on the syntactic representations of Lexical-Functional Grammar (Kaplan & Bresnan 1982). A DOP model provides linguistic representations for an unlimited set of sentences by generalizing from a given corpus of annotated exemplars. It operates by decomposing the given representations into (arbitrarily large) fragments and recomposing those pieces to analyze new sentences. The occurrence-frequencies of the fragments are used to determine the most probable analysis of a sentence.

So far, DOP models have been implemented for phrase-structure trees and logical-semantic representations (cf. Bod 1993, 98; Sima'an 1995, 99; Bonnema et al. 1997; Goodman 1998). However, these DOP models are limited in that they cannot account for underlying syntactic and semantic dependencies that are not reflected directly in a surface tree. DOP models for a number of richer representations have been explored (van den Berg et al. 1994; Tugwell 1995), but these approaches have remained context-free in their generative power. In contrast, Lexical-Functional Grammar (Kaplan & Bresnan 1982) is known to be beyond context-free. In Bod & Kaplan (1998), a first DOP model was proposed based on representations defined by LFG theory ("LFG-DOP").[1] This model was studied from a mathematical perspective by Cormons (1999) who also accomplished a first simple experiment with LFG-DOP. Next, Way (1999) studied LFG-DOP as an architecture for machine translation. The current paper contains the first extensive empirical evaluation of LFG-DOP on the currently available LFG-annotated corpora: the Verbmobil corpus and the Homecentre corpus. Both corpora were annotated at Xerox PARC.

Our parser uses fragments from LFG-annotated sentences to parse new sentences, and Monte Carlo techniques to compute the most probable parse. Although our main goal is to test Bod & Kaplan's LFG-DOP model, we will also test a modified version of LFG-DOP which uses a different model for computing fragment probabilities. While Bod & Kaplan treat all fragments probabilistically equal regardless whether they contain generalized features, we will propose a more fine-grained probability model which treats fragments with generalized features as previously unseen events and assigns probabilities to these fragments by means of discounting. The experiments indicate that our probability model outperforms Bod & Kaplan's probability model on the Verbmobil and Homecentre corpora.

The rest of this paper is organized as follows: we first summarize the LFG-DOP model and go into our proposed extension. Next, we explain the Monte Carlo parsing technique for estimating the most probable LFG-parse of a sentence. In section 3, we test our parser on sentences from the LFG-annotated corpora.

2 Summary of LFG-DOP and an Extension

In accordance with Bod (1998), a particular DOP model is described by specifying settings for the following four parameters:

- a formal definition of a well-formed *representation for utterance analyses,*
- a set of *decomposition operations* that divide a given utterance analysis into a set of *fragments,*
- a set of *composition operations* by which such fragments may be recombined to derive an analysis of a new utterance, and
- a *probability model* that indicates how the probability of a new utterance analysis is computed.

[1] DOP models have recently also been proposed for Tree-Adjoining Grammar and Head-driven Phrase Structure Grammar (cf. Neumann & Flickinger 1999).

In defining a DOP model for Lexical-Functional Grammar representations, Bod & Kaplan (1998) give the following settings for DOP's four parameters.

2.1 Representations

The representations used by LFG-DOP are directly taken from LFG: they consist of a c-structure, an f-structure and a mapping ϕ between them (see Kaplan & Bresnan 1982). The following figure shows an example representation for the utterance *Kim eats*. (We leave out some features to keep the example simple.)

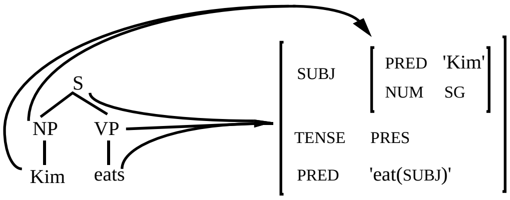


Figure 1. A representation for *Kim eats*

Bod & Kaplan also introduce the notion of accessibility which they later use for defining the decomposition operations of LFG-DOP:

> An f-structure unit *f* is ϕ-*accessible* from a node *n* iff either *n* is ϕ-linked to *f* (that is, *f* = ϕ(*n*)) or *f* is contained within ϕ(*n*) (that is, there is a chain of attributes that leads from ϕ(*n*) to *f*).

According to the LFG representation theory, c-structures and f-structures must satisfy certain formal well-formedness conditions. A c-structure/f-structure pair is a *valid* LFG representation only if it satisfies the Nonbranching Dominance, Uniqueness, Coherence and Completeness conditions (see Kaplan & Bresnan 1982).

2.2 Decomposition operations and Fragments

The fragments for LFG-DOP consist of connected subtrees whose nodes are in ϕ-correspondence with the correponding sub-units of f-structures. To give a precise definition of LFG-DOP fragments, it is convenient to recall the decomposition operations employed by the simpler "Tree-DOP" model which is based on phrase-structure trees only (Bod 1998):

(1) *Root*: the *Root* operation selects any node of a tree to be the root of the new subtree and erases all nodes except the selected node and the nodes it dominates.

(2) *Frontier*: the *Frontier* operation then chooses a set (possibly empty) of nodes in the new subtree different from its root and erases all subtrees dominated by the chosen nodes.

Bod & Kaplan extend Tree-DOP's *Root* and *Frontier* operations so that they also apply to the nodes of the c-structure in LFG, while respecting the fundamental principles of c-structure/f-structure correspondence.

When a node is selected by the *Root* operation, all nodes outside of that node's subtree are erased, just as in Tree-DOP. Further, for LFG-DOP, all ϕ links leaving the erased nodes are removed and all f-structure units that are not ϕ-accessible from the remaining nodes are erased. For example, if *Root* selects the NP in figure 1, then the f-structure corresponding to the S node is erased, giving figure 2 as a possible fragment:

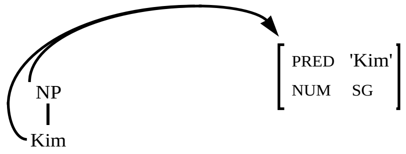


Figure 2. An LFG-DOP fragment obtained by *Root*

In addition the *Root* operation deletes from the remaining f-structure all semantic forms that are local to f-structures that correspond to erased c-structure nodes, and it thereby also maintains the fundamental two-way connection between words and meanings. Thus, if *Root* selects the VP node so that the NP is erased, the subject semantic form "Kim" is also deleted:

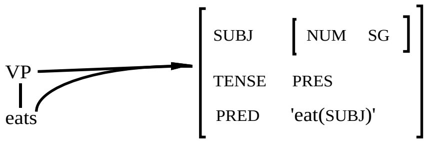


Figure 3. Another LFG-DOP fragment

As with Tree-DOP, the *Frontier* operation then selects a set of frontier nodes and deletes all subtrees they dominate. Like *Root*, it also removes the ϕ links of the deleted nodes and erases any semantic form that corresponds to any of those nodes. *Frontier* does not delete any other f-structure features, however. For instance, if the NP in figure 1 is selected as a frontier node, *Frontier* erases the predicate "Kim" from the fragment:

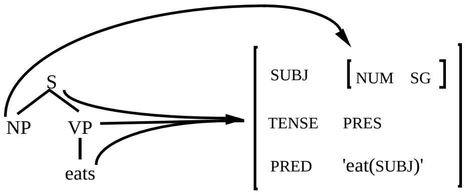


Figure 4. A *Frontier*-generated fragment

Finally, Bod & Kaplan present a third decomposition operation, *Discard*, defined to construct generalizations of the fragments supplied by *Root* and *Frontier*. *Discard* acts to delete combinations of attribute-value pairs subject to the following condition: *Discard* does not delete pairs whose values ϕ-correspond to remaining c-structure nodes. *Discard* produces fragments such as in figure 5, where the subject's number in figure 3 has been deleted:

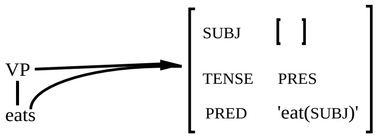


Figure 5. A *Discard*-generated fragment

2.3 The composition operation

In LFG-DOP the operation for combining fragments, indicated by ∘, is carried out in two steps. First the c-structures are combined by left-most substitution subject to the category-matching condition, just as in Tree-DOP (cf. Bod 1993, 98). This is followed by the recursive unification of the f-structures corresponding to the matching nodes. A derivation for an LFG-DOP representation *R* is a sequence of fragments the first of which is labeled with S and for which the iterative application of the composition operation produces *R*.

The two-stage composition operation is illustrated by a simple example. We therefore assume a corpus containing the representation in figure 1 for the sentence *Kim eats* and the representation in figure 6 for the sentence *John fell*.

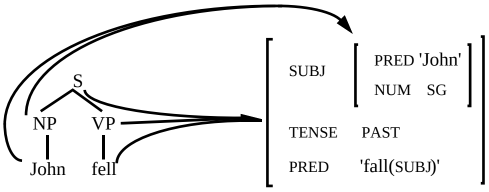


Figure 6. Corpus representation for *John fell*

Figure 7 shows the effect of the LFG-DOP composition operation using two fragments from this corpus, resulting in a representation for the new sentence *Kim fell*.

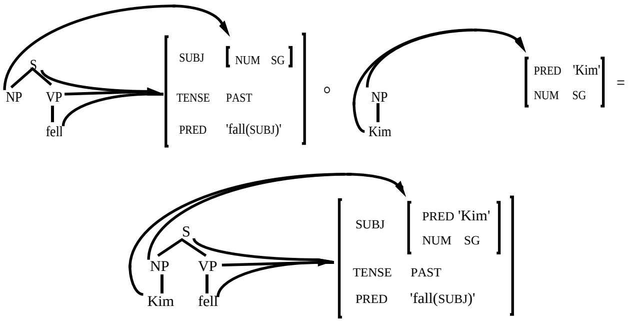


Figure 7. Illustration of the composition operation

This representation satisfies the well-formedness conditions and is therefore valid. Note that the sentence *Kim fell* can be parsed by fragments that are generated by the decomposition operations *Root* and *Frontier* only, without using generalized fragments (i.e. fragments generated by the *Discard* operation). Bod & Kaplan (1998) call a sentence "grammatical with respect to a corpus" if it can be parsed without generalized fragments. Generalized fragments are needed only to parse sentences that are "ungrammatical with respect to the corpus".

2.4 Probability models

As in Tree-DOP, an LFG-DOP representation *R* can typically be derived in many different ways. If each derivation *D* has a probability P(*D*), then the probability of deriving *R* is the sum of the individual derivation probabilities, as shown in (1):

(1) $$P(R) = \Sigma_{D \text{ derives } R}\, P(D)$$

An LFG-DOP derivation is produced by a stochastic process which starts by randomly choosing a fragment whose c-structure is labeled with the initial category (e.g. S). At each subsequent step, a next fragment is chosen at random from among the fragments that can be composed with the current subanalysis. The chosen fragment is composed with the current subanalysis to produce a new one; the process stops when an analysis results with no non-terminal leaves. We will call the set of composable fragments at a certain step in the stochastic process the *competition set* at that step. Let CP(f | CS) denote the probability of choosing a fragment f from a competition set CS containing f, then the probability of a derivation $D = <f_1, f_2 \ldots f_k>$ is

(2) $$P(<f_1, f_2 \ldots f_k>) = \Pi_i\, CP(f_i \mid CS_i)$$

where the *competition probability* CP(f | CS) is expressed in terms of fragment probabilities P(f):

(3) $$CP(f \mid CS) = \frac{P(f)}{\Sigma_{f' \in CS}\, P(f')}$$

Bod & Kaplan give three definitions of increasing complexity for the competition set: the first definition groups all fragments that only satisfy the Category-matching condition of the composition operation (thus leaving out the Uniqueness, Coherence and Completeness conditions); the second definition groups all fragments which satisfy both Category-matching and Uniqueness; and the third definition groups all fragments which satisfy Category-matching, Uniqueness and Coherence. Bod & Kaplan point out that the Completeness condition cannot be enforced at each step of the stochastic derivation process. It is a property of the final representation which can only be enforced by sampling valid representations from the output of the stochastic process.

In this paper, we will only deal with the third definition of competition set, as it selects only those fragments at each derivation step that may finally result in a valid LFG representation, thus reducing the off-line validity checking just to the Completeness condition.

Notice that the computation of the competition probability in (3) still requires a definition for the fragment probability P(*f*). Bod & Kaplan define the probability of a fragment simply as its relative frequency in the bag of all fragments generated from the corpus. Thus Bod & Kaplan do not distinguish between *Root*/*Frontier*-generated fragments and *Discard*-generated fragments, the latter being generalizations over *Root*/*Frontier*-generated fragments. Although Bod & Kaplan illustrate with a simple example that their probability model exhibits a preference for the most specific representation containing the fewest feature generalizations (mainly because specific representations tend to have more derivations than generalized representations), they do not perform an empirical evaluation of their model. We will assess their model on the LFG-annotated Verbmobil and Homecentre corpora in section 3 of this paper.

However, we will also assess an alternative definition of fragment probability which is a refinement of Bod & Kaplan's model. This definition *does* distinguish between fragments supplied by *Root*/*Frontier* and fragments supplied by *Discard*. We will treat the first type of fragments as seen events, and the second type of fragments as previously unseen events. We thus create two separate bags corresponding to two separate distributions: a bag with fragments generated by *Root* and *Frontier*, and a bag with fragments generated by *Discard*. We assign probability mass to the fragments of each bag by means of *discounting*: the relative frequencies of seen events are discounted and the gained probability mass is reserved for the bag of unseen events (cf. Ney et al. 1997). We accomplish this by a very simple estimator: the Turing-Good estimator (Good 1953) which computes the probability mass of unseen events as n_1/N where n_1 is the number of singleton events and N is the total number of seen events. This probability mass is assigned to the bag of *Discard*-generated fragments. The remaining mass $(1 - n_1/N)$ is assigned to the bag of *Root*/*Frontier*-generated fragments. Thus the total probability mass is redistributed over the seen and unseen fragments. The probability of each fragment is then computed as its relative frequency[2] in its bag multiplied by the probability mass assigned to this bag. Let $|f|$ denote the frequency of a fragment *f*, then its probability is given by:

(4) $$P(f \mid f \text{ is generated by } Root/Frontier) =$$

$$(1 - n_1/N)\ \frac{|f|}{\Sigma_{f':\ f' \text{ is generated by } Root/Frontier}\ |f'|}$$

(5) $$P(f \mid f \text{ is generated by } Discard) =$$

$$(n_1/N)\ \frac{|f|}{\Sigma_{f':\ f' \text{ is generated by } Discard}\ |f'|}$$

Note that this probability model assigns less probability mass to *Discard*-generated fragments than Bod & Kaplan's model. For each *Root*/*Frontier*-generated fragment there are exponentially many *Discard*-generated fragments (exponential in the number of features the fragment contains), which means that in Bod & Kaplan's model the *Discard*-generated fragments absorb a vast amount of probability mass. Our model, on the other hand, assigns a fixed probability mass to the distribution of *Discard*-generated fragments and therefore the exponential explosion of these fragments does not affect the probabilities of *Root*/*Frontier*-generated fragments.

3 Testing the LFG-DOP model

3.1 Computing the most probable analysis

In his PhD-thesis, Cormons (1999) describes a parsing algorithm for LFG-DOP which is based on the Tree-DOP parsing technique given in Bod (1998). Cormons first converts LFG-representations into more compact indexed trees: each node in the c-structure is assigned an index which refers to the ϕ-corresponding f-structure unit. For example, the representation in figure 6 is indexed as

```
(S.1     (NP.2   John.2)
         (VP.1   fell.1))
```

where

```
1 --> [  (SUBJ = 2)
         (TENSE = PAST)
         (PRED = fall(SUBJ)) ]

2 --> [ (PRED = John)
        (NUM = SG) ]
```

The indexed trees are then fragmented by applying the Tree-DOP decomposition operations described in section 2. Next, the LFG-DOP decomposition operations *Root*, *Frontier* and *Discard* are applied to the f-structure units that correspond to the indices in the c-structure subtrees. Having obtained the set of LFG-DOP fragments in this way, each test sentence is parsed by a bottom-up chart parser using initially the indexed subtrees only. Thus only the Category-matching condition is enforced during

[2] Bod (2000) discusses some alternative fragment probability estimators, e.g. based on maximum likelihood.

the chart-parsing process. The Uniqueness and Coherence conditions of the corresponding f-structure units are enforced during the disambiguation (or chart-decoding) process. Disambiguation is accomplished by computing a large number of random derivations from the chart; this technique is known as "Monte Carlo disambiguation" and has been extensively described in the literature (e.g. Bod 1998; Chappelier & Rajman 1998; Goodman 1998). Sampling a random derivation from the chart consists of choosing at random one of the fragments from the set of composable fragments at every labeled chart-entry (in a top-down, leftmost order so as to maintain the LFG-DOP derivation order). Thus the competition set of composable fragments is computed on the fly at each derivation step during the Monte Carlo sampling process by grouping the f-structure units that unify and that are coherent with the subderivation built so far.

As mentioned in 2.4, the Completeness condition can only be checked after the derivation process. Incomplete derivations are simply removed from the sampling distribution. After sampling a large number of random derivations that satisfy the LFG validity requirements, the most probable analysis is estimated by the analysis which results most often from the sampled derivations. For our experiments in section 3.2, we used a sample size of N = 10,000 derivations which corresponds to a maximal standard error σ of 0.005 ($\sigma \leq 1/(2\sqrt{N})$, see Bod 1998).

3.2 Experiments with LFG-DOP

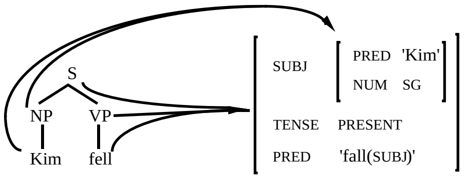
We tested LFG-DOP on two LFG-annotated corpora: the Verbmobil corpus, which contains appointment planning dialogues, and the Homecentre corpus, which contains Xerox printer documentation. Both corpora have been annotated by Xerox PARC. They contain packed LFG-representations (Maxwell & Kaplan 1991) of the grammatical parses of each sentence together with an indication which of these parses is the correct one. The parses are represented in a binary form and were debinarized using software provided to us by Xerox PARC.[3] For our experiments we only used the correct parses of each sentence resulting in 540 Verbmobil parses and 980 Homecentre parses. Each corpus was divided into a 90% training set and a 10% test set. This division was random except for one constraint: that all the words in the test set actually occurred in the training set. The sentences from the test set were parsed and disambiguated by means of the fragments from the training set. Due to memory limitations, we limited the depth of the indexed subtrees to 4. Because of the small size of the corpora we averaged our results on 10 different training/test set splits. Besides an *exact match* accuracy metric, we also used a more fine-grained metric based on the well-known PARSEVAL metrics that evaluate phrase-structure trees (Black et al. 1991). The PARSEVAL metrics compare a proposed parse P with the corresponding correct treebank parse T as follows:

$$\text{Precision} = \frac{\text{\# correct constituents in } P}{\text{\# constituents in } P}$$

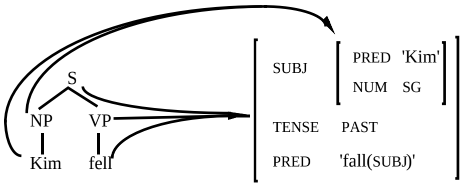
$$\text{Recall} = \frac{\text{\# correct constituents in } P}{\text{\# constituents in } T}$$

In order to apply these metrics to LFG analyses, we extend the PARSEVAL notion of "correct constituent" in the following way: a constituent in P is correct if there exists a constituent in T of the same label that spans the same words and that ϕ-corresponds to the same f-structure unit.

We illustrate the evaluation metrics with a simple example. In the next figure, a proposed parse P is compared with the correct parse T for the test sentence *Kim fell*. The proposed parse is incorrect since it has the incorrect feature value for the TENSE attribute. Thus, if this were the only test sentence, the exact match would be 0%. The precision, on the other hand, is higher than 0% as it compares the parse on a constituent basis. Both the proposed parse and the correct parse contain three constituents: S, NP and VP. While all three constituents in P have the same label and span the same words as in T, only the NP constituent in P also maps to the same f-structure unit as in T. The precision is thus equal to 1/3. Note that in this example the recall is equal to the precision, but this need not always be the case.



Proposed parse P



Correct parse T

In our experiments we are first of all interested in comparing the performance of Bod & Kaplan's probability model against our probability model (as explained in section 2.4). Moreover, we also want to

[3] Thanks to Hadar Shemtov for providing us with the relevant software.

study the contribution of *Discard*-generated fragments to the parse accuracy. We therefore created for each training set two sets of fragments: one which contains *all* fragments (up to depth 4) and one which excludes the fragments generated by *Discard*. The exclusion of the *Discard*-generated fragments means that all probability mass goes to the fragments generated by *Root* and *Frontier* in which case our model is equivalent to Bod & Kaplan's. The following two tables present the results of our experiments where +Discard refers to the full set of fragments and –Discard refers to the fragment set without *Discard*-generated fragments.

	Exact Match		Precision		Recall	
	+Discard	–Discard	+Discard	–Discard	+Discard	–Discard
Bod&Kaplan98	1.1%	35.2%	13.8%	76.0%	11.5%	74.9%
OurModel	35.9%	35.2%	77.5%	76.0%	76.4%	74.9%

Table 1. Experimental results on the Verbmobil corpus

	Exact Match		Precision		Recall	
	+Discard	–Discard	+Discard	–Discard	+Discard	–Discard
Bod&Kaplan98	2.7%	37.9%	17.1%	77.8%	15.5%	77.2%
OurModel	38.4%	37.9%	80.0%	77.8%	78.6%	77.2%

Table 2. Experimental results on the Homecentre corpus

The tables show that Bod & Kaplan's model scores extremely bad if all fragments are used: the exact match is only 1.1% on the Verbmobil corpus and 2.7% on the Homecentre corpus, whereas our model scores respectively 35.9% and 38.4% on these corpora. Also the more fine-grained precision and recall scores of Bod & Kaplan's model are quite low: e.g. 13.8% and 11.5% on the Verbmobil corpus, where our model obtains 77.5% and 76.4%. We found out that even for the few test sentences that occur literally in the training set, Bod & Kaplan's model does not always generate the correct analysis, whereas our model does. Interestingly, the accuracy of Bod & Kaplan's model is much higher if *Discard*-generated fragments are excluded. This suggests that treating generalized fragments probabilistically in the same way as ungeneralized fragments is harmful. Cormons (1999) has made a mathematical observation which also shows that generalized fragments can get too much probability mass.

The tables also show that our way of assigning probabilities to *Discard*-generated fragments leads only to a slight accuracy increase (compared to the experiments in which *Discard*-generated fragments are excluded). According to paired *t*-testing none of these differences in accuracy were statistically significant. This suggests that *Discard*-generated fragments do not significantly contribute to the parse accuracy, or that perhaps these fragments are too numerous to be reliably estimated on the basis of our small corpora. We also varied the probability mass assigned to *Discard*-generated fragments: except for very small (≤ 0.01) or large values (≥ 0.88), which led to an accuracy decrease, there was no significant change.[4]

It is difficult to say how good or bad our results are with respect to other approaches. The only other published results on the LFG-annotated Verbmobil and Homecentre corpora are by Johnson et al. (1999) and Johnson & Riezler (2000) who use a log-linear model to estimate probabilities. But while we first parse the test sentences with fragments from the training set and subsequently compute the most probable parse, Johnson et al. directly use the packed LFG-representations from the test set to select the most probable parse, thereby completely skipping the parsing phase (Mark Johnson, p.c.). Moreover, 42% of the Verbmobil sentences and 51% of the Homecentre sentences are unambiguous (i.e. their packed LFG-representations contain only one analysis), which makes Johnson et al's task completely trivial for these sentences. In our approach, *all* test sentences were ambiguous, resulting in a much more difficult task. A quantitative comparison between our model and Johnson et al.'s is therefore meaningless.

Finally, we are interested in the impact of functional structures on predicting the correct constituent structures. We therefore removed all f-structure units from the fragments (thus yielding a Tree-DOP model) and compared the results against our version of LFG-DOP (which include the *Discard*-generated fragments). We evaluated the parse accuracy on the tree-structures only, using exact match together with the PARSEVAL measures. We used the same training/test set splits as in the previous experiments and limited the maximum subtree depth again to 4. The following tables show the results.

	Exact Match	Precision	Recall
Tree-DOP	46.6%	88.9%	86.7%
LFG-DOP	50.8%	90.3%	88.4%

Table 3. C-structure accuracy on the Verbmobil

[4] Although generalized fragments thus seem statistically unimportant for these corpora, they remain important for parsing ungrammatical sentences (which was the original motivation for including them -- see Bod & Kaplan 1998).

	Exact Match	Precision	Recall
Tree-DOP	49.0%	93.4%	92.1%
LFG-DOP	53.2%	95.8%	94.7%

Table 4. C-structure accuracy on the Homecentre

The results indicate that LFG-DOP's functional structures help to improve the parse accuracy of tree-structures. In other words, LFG-DOP outperforms Tree-DOP if evaluated on tree-structures only. According to paired *t*-tests the differences in accuracy were statistically significant.

4 Conclusion

We have given an empirical assessment of the LFG-DOP model introduced by Bod & Kaplan (1998). We developed a new probability model for LFG-DOP which treats fragments with generalized features as previously unseen events. The experiments showed that our probability model outperforms Bod & Kaplan's model on the Verbmobil and Homecentre corpora. Moreover, Bod & Kaplan's model turned out to be inadequate in dealing with generalized fragments. We also established that the contribution of generalized fragments to the parse accuracy in our model is minimal and statistically insignificant. Finally, we showed that LFG's functional structures contribute to significantly higher parse accuracy on tree structures. This suggests that our model may be successfully used to exploit the functional annotations in the Penn Treebank (Marcus et al. 1994), provided that these annotations can be converted into LFG-style functional structures. As future research, we want to test LFG-DOP using log-linear models, as such models maximize the likelihood of the training corpus.

References

M. van den Berg, R. Bod and R. Scha, 1994. "A Corpus-Based Approach to Semantic Interpretation", *Proceedings Ninth Amsterdam Colloquium*, Amsterdam, The Netherlands.

E. Black et al., 1991. "A Procedure for Quantitatively Comparing the Syntactic Coverage of English", *Proceedings DARPA Speech and Natural Language Workshop*, Pacific Grove, Morgan Kaufmann.

R. Bod, 1993. "Using an Annotated Language Corpus as a Virtual Stochastic Grammar", *Proceedings AAAI'93*, Washington D.C.

R. Bod, 1998. *Beyond Grammar*, CSLI Publications, Cambridge University Press.

R. Bod, 2000. "Parsing with the Shortest Derivation", *Proceedings COLING-2000*, Saarbrücken, Germany.

R. Bod and R. Kaplan, 1998. "A Probabilistic Corpus-Driven Model for Lexical Functional Analysis", *Proceedings COLING-ACL'98*, Montreal, Canada.

R. Bonnema, R. Bod and R. Scha, 1997. "A DOP Model for Semantic Interpretation", *Proceedings ACL/EACL-97*, Madrid, Spain.

J. Chappelier and M. Rajman, 1998. "Extraction stochastique d'arbres d'analyse pour le modèle DOP", *Proceedings TALN'98*, Paris, France.

B. Cormons, 1999. *Analyse et désambiguisation: Une approche à base de corpus (Data-Oriented Parsing) pour les répresentations lexicales fonctionnelles*. PhD thesis, Université de Rennes, France.

I. Good, 1953. "The Population Frequencies of Species and the Estimation of Population Parameters", *Biometrika* 40, 237-264.

J. Goodman, 1998. *Parsing Inside-Out*, PhD thesis, Harvard University, Mass.

M. Johnson, S. Geman, S. Canon, Z. Chi and S. Riezler, 1999. "Estimators for Stochastic Unification-Based Grammars", *Proceedings ACL'99*, Maryland.

M. Johnson and S. Riezler, 2000. "Exploiting Auxiliary Distributions in Stochastic Unification-Based Grammars", *Proceedings ANLP-NAACL-2000*, Seattle, Washington.

R. Kaplan, and J. Bresnan, 1982. "Lexical-Functional Grammar: A Formal System for Grammatical Representation", in J. Bresnan (ed.), *The Mental Representation of Grammatical Relations*, The MIT Press, Cambridge, Mass.

M. Marcus, G. Kim, M. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz and B. Schasberger, 1994. "The Penn Treebank: Annotating Predicate Argument Structure". In: *ARPA Human Language Technology Workshop*, 110-115.

J. Maxwell and R. Kaplan, 1991. "A Method for Disjunctive Constraint Satisfaction", in M. Tomita (ed.), *Current Issues in Parsing Technology*, Kluwer Academic Publishers.

G. Neumann and D. Flickinger, 1999. "Learning Stochastic Lexicalized Tree Grammars from HPSG", DFKI Technical Report, Saarbrücken, Germany.

H. Ney, S. Martin and F. Wessel, 1997. "Statistical Language Modeling Using Leaving-One-Out", in S. Young & G. Bloothooft (eds.), *Corpus-Based Methods in Language and Speech Processing*, Kluwer Academic Publishers.

K. Sima'an, 1995. "An optimized algorithm for Data Oriented Parsing", in R. Mitkov and N. Nicolov (eds.), *Recent Advances in Natural Language Processing 1995*, volume 136 of *Current Issues in Linguistic Theory*. John Benjamins, Amsterdam.

K. Sima'an, 1999. *Learning Efficient Disambiguation*. PhD thesis, ILLC dissertation series number 1999-02. Utrecht / Amsterdam.

D. Tugwell, 1995. "A State-Transition Grammar for Data-Oriented Parsing", *Proceedings European Chapter of the ACL'95*, Dublin, Ireland.

A. Way, 1999. "A Hybrid Architecture for Robust MT using LFG-DOP", *Journal of Experimental and Theoretical Artificial Intelligence* 11 (Special Issue on Memory-Based Language Processing)